Rapport_Projet

Armel SOUBEIGA

Sunday, December 6, 2017

Construction du package

Description

popsize

Elle est construite à partir d'une fonction appellée *estimation*. Cette fonction conient des bout de code S3 et des données de type *.RData* compilé avec. Ces données seront utilisées dans la fonction du package et dans le *shiny* lorsqu'on appellera la library.

```
#' @title Estimation de la taille de population
#' @description Prend en parametres des coordonnées géographique et retour la taille de population à ce
#' @return la taille de population
#' @param Y0
#' @param X0
#' @export
#'
#'
estimation <- function(X0,Y0)
{

  data(MyData)
  pop <- base[,17]
  #matrice de distance
  cords<-base[c("POINT_X","POINT_Y")]
  S0<-c(X0,Y0)
  cords<-as.matrix(rbind(S0,cords))
  dist <- as.matrix(dist(cords))

  #Fonction S3
  MatrixCor<- function(x,pa,ep, maybe = "some", other = "arguments", ...) {
    UseMethod("MatrixCor")
  }

  MatrixCor.matrix<-function(x,pa,ep)
  {
    po=mean(x)

    for( i in 1:length(x)){

      dist1 <- x

      if(dist1[i]==0){
        dist1[i]=0
      }
```

```
      else if(dist1[i] > 0 & dist1[i] <= po){
        dist1[i]= ep+(pa-ep)*(1.5*(dist1[i]/po)-0.5*(dist1[i]/po)^3)
      }

      else if(dist1[i]> po){}
      dist1[i]=pa
    }
    return(dist1)
  }
  pa=4593.973
  ep=2267.514
  #covariance
  dist2=MatrixCor(dist,pa,ep)
  for( i in 1:length(dist2))
    dist2[i]<- (pa-dist2[i])

  #calcul des coeficient (K*coef=k0)
  #K
  K1<- dist2[-1,-1]
  K2<- cbind(K1,1)
  v1<-rep(1,length(K1[,1]))
  v2<-c(v1,0)
  K<-rbind(K2,v2)

  #k0
  kk<-dist2[1,]
  k0<-c(kk[-1],1)

  #coef
  coef<-k0%*%solve(K)
  coefs<-coef[-length(coef)]
  Z0<-sum(coefs%*%pop)

  #resultat
  return(Z0)
}
```

Exemple

Il faut dispose de coordonnées géographique en utm de la zone d'étude

```
library("xtable")
library("gstat")
library("rgdal")
```

```
## Loading required package: sp
```

```
## rgdal: version: 1.1-8, (SVN revision 616)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.0.1, released 2015/09/15
##  Path to GDAL shared files: C:/Users/Armel soubeiga/Documents/R/win-library/3.1/rgdal/gdal
##  GDAL does not use iconv for recoding strings.
```

```
##  Loaded PROJ.4 runtime: Rel. 4.9.1, 04 March 2015, [PJ_VERSION: 491]
##  Path to PROJ.4 shared files: C:/Users/Armel soubeiga/Documents/R/win-library/3.1/rgdal/proj
##  Linking to sp version: 1.2-3
```

```
library("sp")
library("maptools")
```

```
## Checking rgeos availability: TRUE
```

```
##
## Attaching package: 'maptools'
```

```
## The following object is masked from 'package:xtable':
##
##     label
```

```
library('Rcpp')
library("popsize")
```

```
##
## Attaching package: 'popsize'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     estimation
```

```
data(MyData)
Base <- base
head(Base)
```

```
##   Arr_ Secteur Id_lieudit       DATE Longitude Laltitude  POINT_X POINT_Y
## 1    5      17          1 30/05/2016  -4.26936  11.15995 361395.0 1233962
## 2    5      17          2 30/05/2016  -4.26902  11.15893 361431.6 1233850
## 3    5      17          3 30/05/2016  -4.26761  11.17169 361591.7 1235260
## 4    5       5          4 30/05/2016  -4.27942  11.16170 360297.2 1234161
## 5    5      17          5 30/05/2016  -4.28078  11.15954 360147.6 1233923
## 6    5       5          6 30/05/2016  -4.28213  11.15270 359996.9 1233167
##    Type_ZONE Precision                   Noms Code Nb_parcel
## 1 Habitation         8        csps de sarfalao    2        20
## 2 Habitation         8       pharmacie-remedis    1        33
## 3 Habitation         8        pharmacie hereso    1        19
## 4 Habitation         8 mairie arrondissement 5    3        10
## 5 Habitation         8      pharmacie sarfalao    1        35
## 6 Habitation         8      pharmacie la grace    1        22
##   Nb_parcel_NonConst Nb_parcel_Const  Menage      pop
## 1                  3              17 22.4485 116.7322
## 2                  0              33 43.5765 226.5978
## 3                  0              19 25.0895 130.4654
## 4                  1               9 11.8845  61.7994
## 5                  0              35 46.2175 240.3310
## 6                  1              21 27.7305 144.1986
```

```
estimation(362889.041656,1238600.13535)
```

```
## [1] 161.2407
```

shiny

Dans le server du shiny on retrouve quelques petite fonction simple et d'autre faite avec du *RCPP*

Cartographie

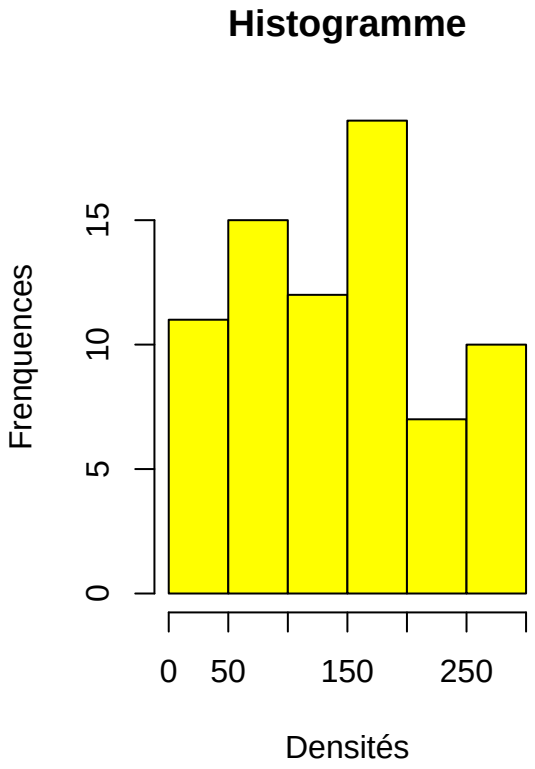
```
comune<-readShapePoly("C:\\Users\\Armel soubeiga\\Desktop\\Mes COURS\\SSD_UGA\\M1\\R\\ProjetR_Armel\\SI
 par(mar=c(0, 0, 0, 0))
 plot(comune[comune$NOMDEP=="BOBO-DIOULASSO",],col="gray")
 points(362889.041656,1238600.13535,col="red",pch=20, cex=1)
```

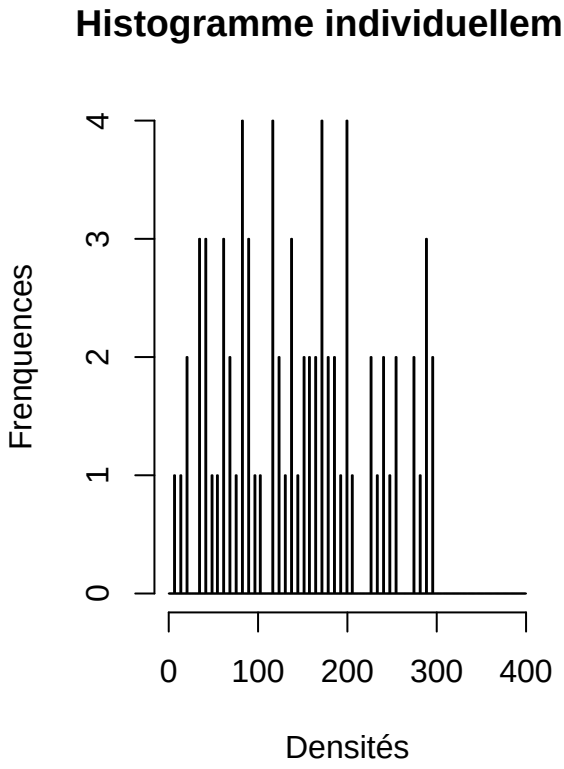
Statistique

```
as.matrix(summary(Base$pop))
```

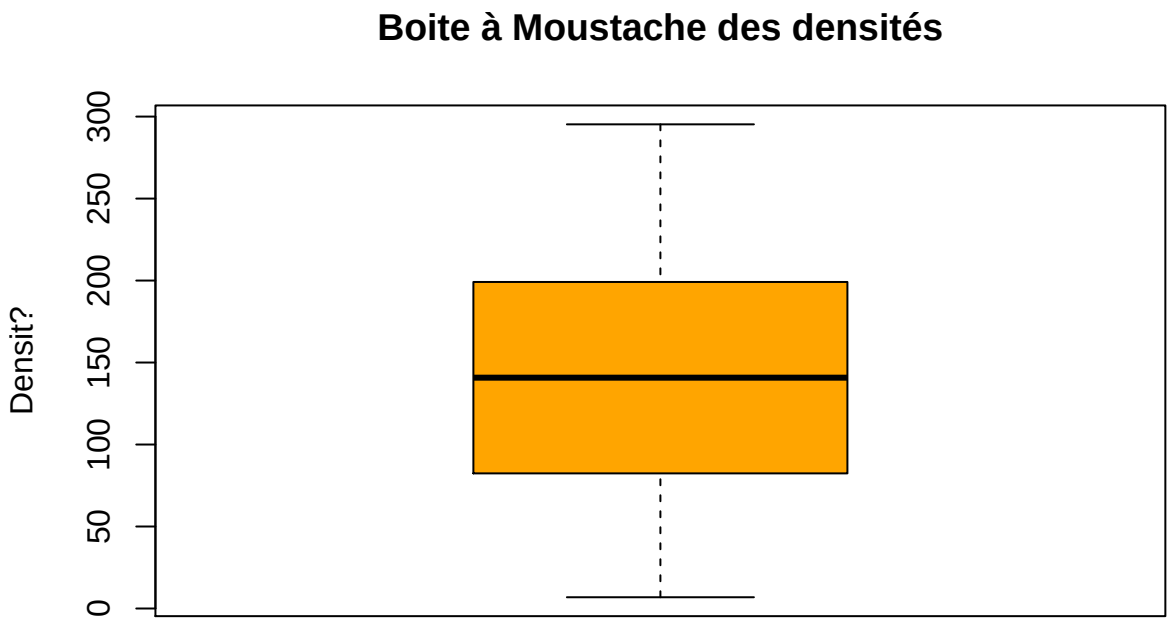
```
##            [,1]
## Min.      6.867
## 1st Qu.  82.400
## Median  140.800
## Mean    145.300
## 3rd Qu. 199.100
## Max.    295.300
```

```
par(mfrow=c(1,2))
hist(Base$pop,col="yellow",main="Histogramme",xlab="Densités",ylab="Frenquences")
hist(Base$pop,breaks=seq(0,400, by=1),main="Histogramme individuellement",xlab="Densités",ylab="Frenqu
```

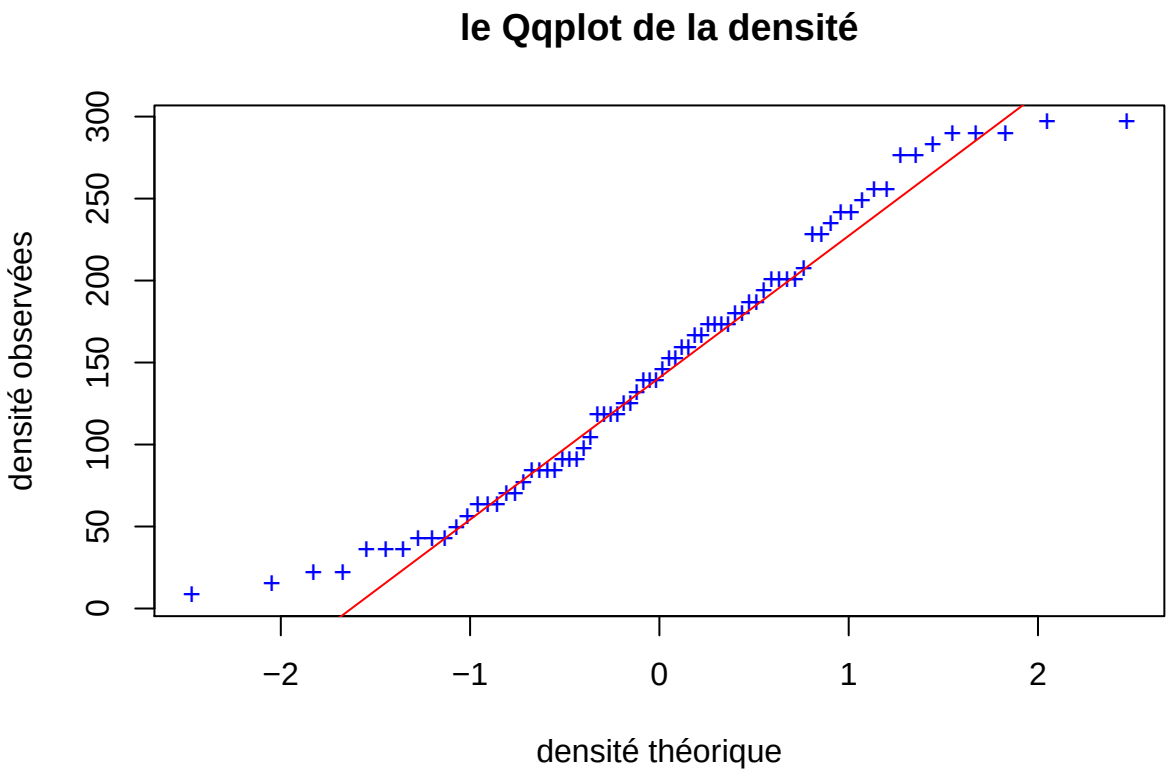





```
boxplot(Base$pop,main="Boite à Moustache des densités",col="orange",ylab="Densit?")
```



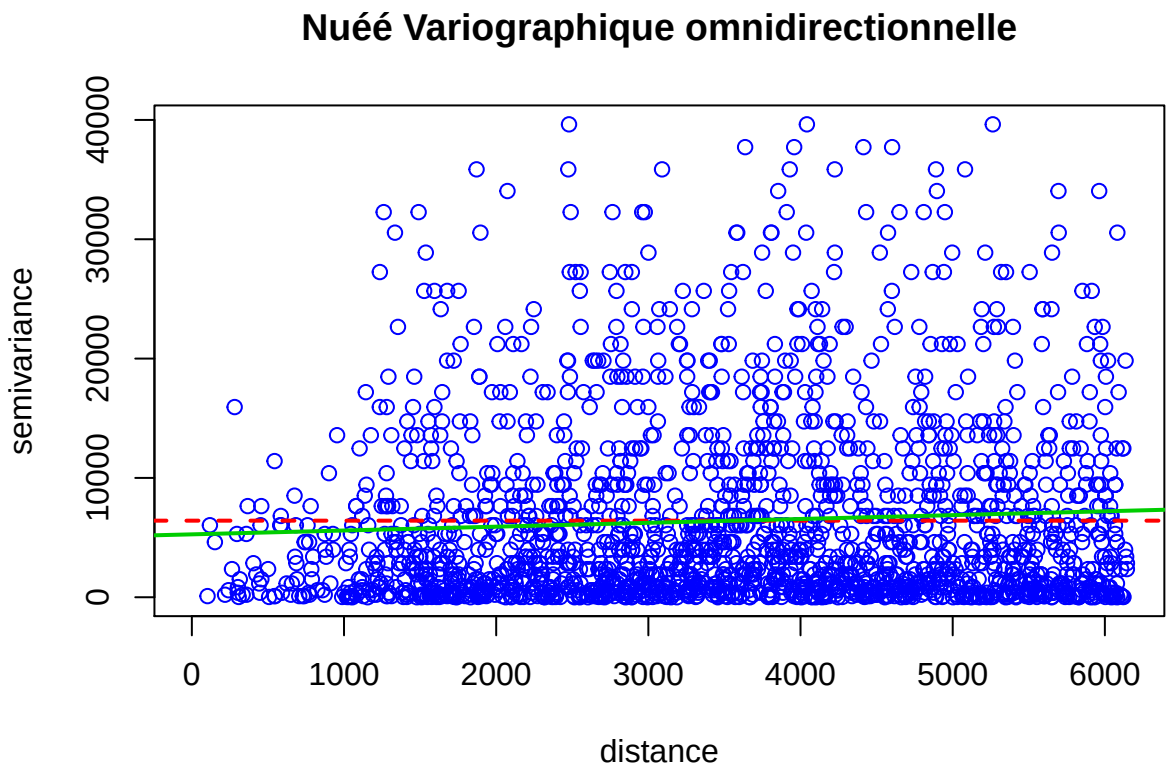

```
qqnorm(Base$pop,pch="+",col="blue",main="le Qqplot de la densité",xlab="densité théorique",ylab="densité
  qqline(Base$pop,col="red")
```



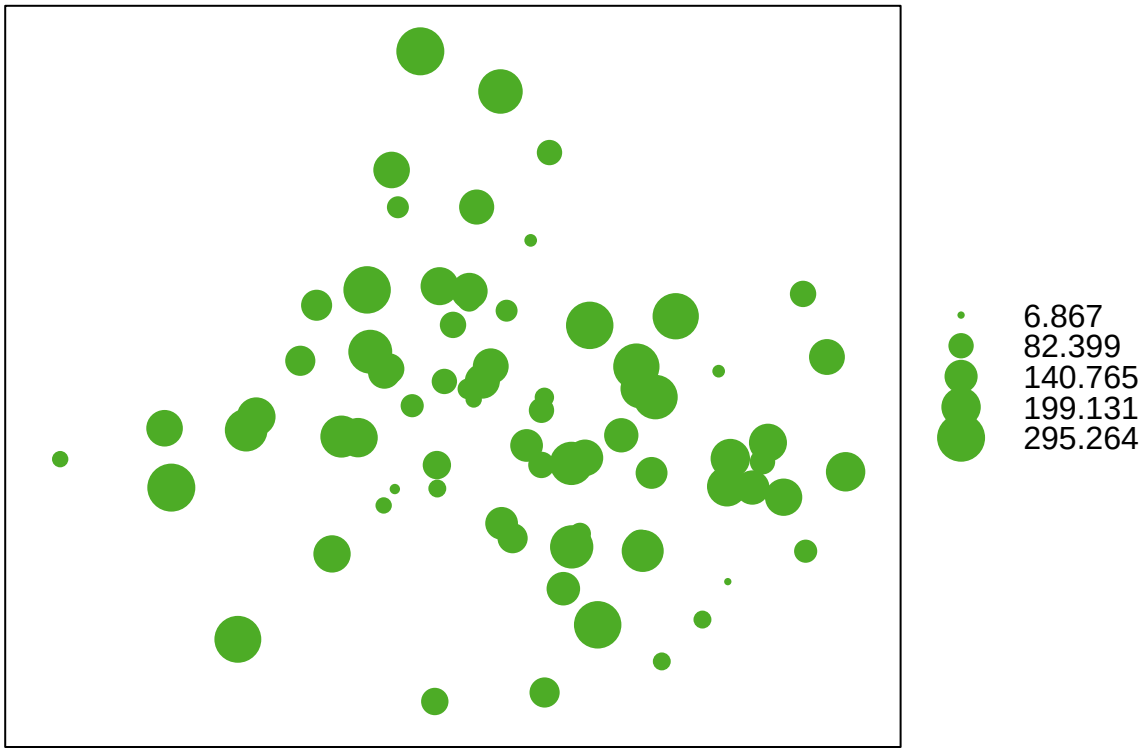

```
coordinates(Base)= ~POINT_X+POINT_Y
 vgm<-variogram(Base$pop~1, Base, cloud=TRUE)
 pp<-plot(vgm,col="blue",main="Nuéé Variographique omnidirectionnelle",id=TRUE)
```

```
## [1] "mouse-left identifies, mouse-right or Esc stops"
```

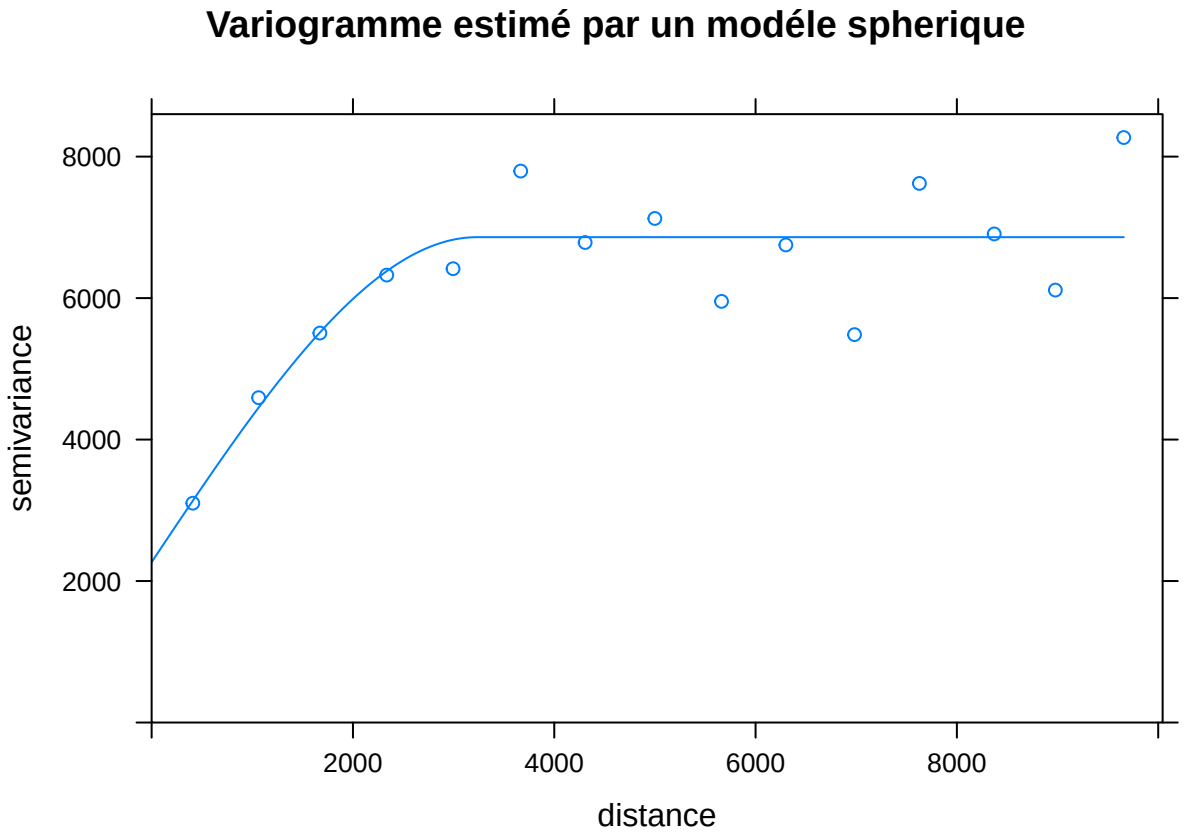
```
 abline(h=mean(vgm$gamma), col="red",lwd=2, lty=2)
 vgmreg <- lm(vgm$gamma~vgm$dist)
 abline(vgmreg, col=3, lwd=2)
```




```
bubble(Base,"pop",main="Dispersion des Lieudits en fonction de la densité")
```




```
pop.vgm<- gstat::variogram(pop~ 1 , Base, cutoff=10000)
pop.fit<-gstat::fit.variogram(pop.vgm,vgm("Sph"))
plot(pop.vgm , pop.fit, main= "Variogramme estimé par un modéle spherique")
```



cppFunction('double sigmapop(NumericVector x) { double sigma = sd(x); return sigma; }')

Mean

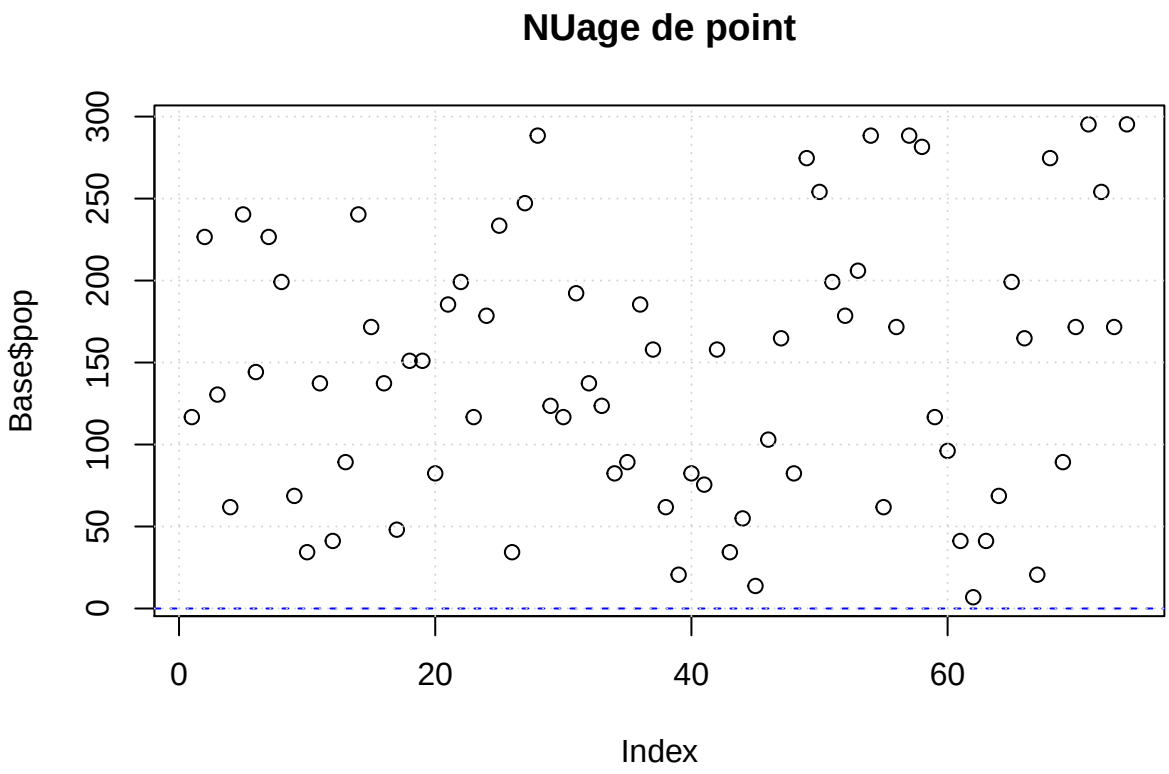
cppFunction('double meanpop(NumericVector x) { int n = x.size(); double total = 0;

```
for(int i = 0; i < n; ++i) {
total += x[i];
}
return total / n;
}')
```

Test normalité

mm <- meanpop(Base$pop)$sd <- sigmapop(Base$pop) ks.test(Base$pop,"pnorm",mm,sd)

```
plot(Base$pop,type="p",
       main=paste("NUage de point"))
  abline(h=0,col="blue",lty=2)
  grid()
```


NUage de point
Base$pop
0
50
100
150
200
250
300
0
20
40
60
Index